

TRUSTED DATA HAULING BY CO-OPERATIVE SPECTRUM SENSING (CSS) IN COGNITIVE RADIO NETWORK

ELANAGAI GUNASEKARAN

Abstract - In CSS, there are many energy efficient traditional methods are their but it doesn't bother about Network security. Hence we proposed an Energy Efficient Collaborative Spectrum Sensing (EECSS) Protocol based on Trust and Reputation Management (TRM) Unit to combat malicious behavior. TRM unit calculates Trust value for each Secondary user's (SU's) and find out and drops them from the network. By this way we prove that network security is enabled in our model. Energy Consumption is the major problem in Traditional CSS method, due to more number of sensing reports exchanged. In our proposed method we are using Data Fusion scheme to reduce total number of sensing reports exchanged between secondary user's base station and SU's i.e. energy consumption was reduced.

Key Words: Cognitive Radio Network, Co-operative Spectrum Sensing, Data Fusion, Trust and Reputation Management Unit, Spectrum Sensing

1. INTRODUCTION

Cognitive Radio (CR) has been proposed as a solution to destroy Radio Spectrum Scarcity. In CRN communication devices changes its transmission and reception parameters for efficient utilization of spectrum this is called as Dynamic Spectrum Access. In CRN, Secondary users (SU) can sense unused licensed spectrum band. Important activity of CRN is to allocate unused spectrum holes to SU without providing interference to Primary users. Spectrum Sensing is the method used to detect the unused spectrum and sharing it. Cooperative spectrum sensing (CSS) has been proposed in which sensing reports are coming up with few decision making authorities to ensure about reliable decision on the state of spectrum usage. However in the presence of malicious SU's integrity of report need to be checked to avoid interference. Hence Trust and Reputation Management unit is used it is not only used to combat the malicious behavior, it ensures that there is no link disconnection in the network. Since the network is coming up with few decision authorities there is no secondary users loose the chance to utilize the spectrum when they are within the range of CRN simultaneously malicious nodes are avoided to sense the spectrum. Including decision making authorities in the cognitive radio network reduces the energy consumption. Reduction of total number of sensing report is our main objective this will be achieved by using the proposed protocol. There are some spectrum sensing techniques are there which are just used to allocate the unused spectrum holes hence we including few decision making authorities in the network to produce the efficacious and trusted communication.

2. SYSTEM MODEL

Here we are using H -Honest secondary users (HSU) and one Primary user base station (PUBS) and secondary user base station (SUBS) and three primary users (PU). Here SUBS and PUBS base stations are communicated so that PUBS conveys State Band Matrix (SBM) which includes the state of the licensed band. Here Fusion Centre (SUBS) requests all SU to sense one band. BSM allows FC to calculate accuracy of its report in addition with the report from SU's. The System model was gonna run in NS2 Software.

3. PROPOSED WORK CODINGS

```
# Define options
set val(chan) Channel/WirelessChannel ;# channel type
set val(prop) Propagation/TwoRayGround ;# radio-
propagation model
set val(netif) Phy/WirelessPhy ;# network interface type
set val(mac) Mac/802_11 ;# MAC type
set val(ifq) Queue/DropTail/PriQueue ;# interface queue
type
set val(ll) LL ;# link layer type
set val(ant) Antenna/OmniAntenna ;# antenna model
set val(ifqlen) 30 ;# max packet in ifq
set val(nn) 15 ;# number of mobilenodes
set val(rp) AODV ;# routing protocol
set val(x) 1000 ;# X dimension of topography
set val(y) 1000 ;# Y dimension of topography
set val(stop) 150 ;# time of simulation end
set ns [new Simulator]
set tracefd [open simple.tr w]
set namtrace [open simwrls.nam w]
$ns trace-all $tracefd
$ns namtrace-all-wireless $namtrace $val(x) $val(y)
# set up topography object
set topo [new Topography]
$topo load_flatgrid $val(x) $val(y)
create-god $val(nn)
```

```

# configure the nodes
$ns node-config -adhocRouting $val(rp) \
-llType $val(ll) \
-macType $val(mac) \
-ifqType $val(ifq) \
-ifqLen $val(ifqlen) \
-antType $val(ant) \
-propType $val(prop) \
-phyType $val(netif) \
-channelType $val(chan) \
-topoInstance $topo \
-agentTrace ON \
-routerTrace ON \
-macTrace OFF \
-movementTrace ON
for {set i 0} {$i < $val(nn)} { incr i } {
set n($i) [$ns node]
}
$n(0) set X_ 10.0
$n(0) set Y_ 15.0
$n(0) set Z_ 0.0
$n(1) set X_ 60.0
$n(1) set Y_ 75.0
$n(1) set Z_ 0.0
$n(2) set X_ 120.0
$n(2) set Y_ 25.0
$n(2) set Z_ 0.0
$n(3) set X_ 180.0
$n(3) set Y_ 95.0
$n(3) set Z_ 0.0
$n(4) set X_ 240.0
$n(4) set Y_ 5.0
$n(4) set Z_ 0.0
$n(5) set X_ 300.0
$n(5) set Y_ 5.0
$n(5) set Z_ 0.0
$n(6) set X_ 360.0
$n(6) set Y_ 125.0
$n(6) set Z_ 0.0
$n(7) set X_ 420.0
$n(7) set Y_ 155.0
$n(7) set Z_ 0.0
$n(8) set X_ 480.0
$n(8) set Y_ 85.0
$n(8) set Z_ 0.0
$n(9) set X_ 540.0
$n(9) set Y_ 225.0
$n(9) set Z_ 0.0
$n(10) set X_ 600.0
$n(10) set Y_ 345.0
$n(10) set Z_ 0.0
$n(11) set X_ 660.0
$n(11) set Y_ 535.0
$n(11) set Z_ 0.0
$n(12) set X_ 720.0
$n(12) set Y_ 705.0
$n(12) set Z_ 0.0
$n(13) set X_ 780.0
$n(13) set Y_ 435.0
$n(13) set Z_ 0.0
$n(14) set X_ 840.0
$n(14) set Y_ 235.0
$n(14) set Z_ 0.0
$ns at 10.0 "$n(12) setdest 600.0 703.0 15.0"
$ns at 11.0 "$n(13) setdest 790.0 710.0 15.0"
$ns at 15.0 "$n(8) setdest 780.0 85.0 15.0"
$ns at 18.0 "$n(10) setdest 780.0 345.0 15.0"
$ns at 20.0 "$n(9) setdest 590.0 223.0 18.0"
$ns at 20.0 "$n(7) setdest 420.0 7.0 16.0"
$ns at 22.0 "$n(4) setdest 6.0 102.0 27.0"
$ns at 23.0 "$n(3) setdest 55.0 152.0 17.0"
# Set a TCP connection and send data between n(4) and
n(1)
set tcp [new Agent/TCP/Newreno]
$tcp set class_ 2
set sink [new Agent/TCPSink]
$ns attach-agent $n(4) $tcp
$ns attach-agent $n(1) $sink

```

```
$ns connect $tcp $sink
set ftp [new Application/FTP]
$ftp attach-agent $tcp
$ns at 25.0 "$ftp start"
$ns at 35.0 "$ftp stop"
# Set a Data between n(1) and n(6)
set tcp [new Agent/TCP/Newreno]
$tcp set class_2
set sink [new Agent/TCPSink]
$ns attach-agent $n(1) $tcp
$ns attach-agent $n(6) $sink
$ns connect $tcp $sink
set ftp [new Application/FTP]
$ftp attach-agent $tcp
$ns at 43.0 "$ftp start"
$ns at 59.0 "$ftp stop"
# Set a Data between n(6) and n(9)
set tcp [new Agent/TCP/Newreno]
$tcp set class_2
set sink [new Agent/TCPSink]
$ns attach-agent $n(6) $tcp
$ns attach-agent $n(9) $sink
$ns connect $tcp $sink
set ftp [new Application/FTP]
$ftp attach-agent $tcp
$ns at 60.0 "$ftp start"
$ns at 85.0 "$ftp stop"
# Send data between n(9) and n(11)
set tcp [new Agent/TCP/Newreno]
$tcp set class_2
set sink [new Agent/TCPSink]
$ns attach-agent $n(9) $tcp
$ns attach-agent $n(11) $sink
$ns connect $tcp $sink
set ftp [new Application/FTP]
$ftp attach-agent $tcp
$ns at 88.0 "$ftp start"
$ns at 100.0 "$ftp stop"

# Set a TCP connection and send data between n(11) and
n(13)
set tcp [new Agent/TCP/Newreno]
$tcp set class_2
set sink [new Agent/TCPSink]
$ns attach-agent $n(11) $tcp
$ns attach-agent $n(13) $sink
$ns connect $tcp $sink
set ftp [new Application/FTP]
$ftp attach-agent $tcp
$ns at 101.0 "$ftp start"
$ns at 116.0 "$ftp stop"
# Define node initial position in nam
$ns initial_node_pos $n(0) 20
$ns initial_node_pos $n(1) 30
$ns initial_node_pos $n(2) 20
$ns initial_node_pos $n(3) 20
$ns initial_node_pos $n(4) 20
$ns initial_node_pos $n(5) 20
$ns initial_node_pos $n(6) 30
$ns initial_node_pos $n(7) 20
$ns initial_node_pos $n(8) 20
$ns initial_node_pos $n(9) 30
$ns initial_node_pos $n(10) 20
$ns initial_node_pos $n(11) 30
$ns initial_node_pos $n(12) 20
$ns initial_node_pos $n(13) 20
$ns initial_node_pos $n(14) 20
# Telling nodes when the simulation ends
for {set i 0} {$i < $val(nn)} {incr i}
{
$ns at $val(stop) "$n($i) reset";
}
# ending nam and the simulation
$ns at $val(stop) "$ns nam-end-wireless $val(stop)"
$ns at $val(stop) "stop"
$ns at 150.1 "puts \"end simulation\" ; $ns halt"
proc stop {} {
global ns tracefd namtrace
```

```
$ns flush-trace
close $tracefd
close $namtrace
exec nam simwrls.nam &
}
$ns run
```

4. CONCLUSION

Empirical Analysis without Malicious node is performed and the result is generated which shows that the Energy consumption in EE-CSS protocol is reduced based on Performance delay which shows that the number of sensing reports exchanged between secondary users and its base station was reduced. Hence without including malicious node we had reduced total number of sensing reports exchanged than traditional one, which shows this proposed method is more energy efficient. This work proves authentication and there is no link breakage in the network hence requested secondary user's spectrum utilization was going correct which will be user-friendly part of our work.

REFERENCES

- [1] [1] Chen.Y (2008) "Optimum number of secondary users in collaborative spectrum sensing considering resource efficiency", IEEE Commun Letter.
- [2] [2] Clansy.T and Goergen.N (2008) "Security in Cognitive Radio Networks: Threats and Mitigation", in Proc.EAI Int.Conf. Crowncom, pp 1-8.
- [3] [3] Duan.L, Zhang.L, Chu.Y and Liu.S (2009) "Cooperative spectrum Sensing with double threshold detection based on reputation", IEEE WiCOM.
- [4] [4] Ghasemi.A and Sousa.E (2007), "Optimization of Collaborative Spectrum Sensing", J Commun,.
- [5] [5] Hong.L, Ma.J, Xu.F, Li.S, and Zhou,Z (2008)"Optimization of collaborative spectrum sensing for cognitive radio," in Proc. IEEE,(ICNSC).
- [6] [6] Li.H and Han.Z (2010) "Catch me if you can: An abnormality detection approach for CSS in Cognitive radio networks", IEEE Transaction, Wireless Commun.